Shira Pahmer  
Final Project- DNS Toolkit Research Portion

Dig command

Dig is a command that preforms DNS lookup, and is basically a more updated version of the nslookup command. It takes in a domain name in the query and returns information about the domain name depending on the query type. With the CAA dig query that is used to retrieve the CAA record of a given domain name, which is the record that contains information about which certificate authorities are allowed to issue certificates for that domain.

When running the command: dig jct.ac.il caa and analyzing the packets in wireshark:

A screenshot of a computer

Description automatically generatedthis is the request packet. It is a dns type packet that is a standard query, requesting the caa record of jct.ac.il domain.

In the response packet, it says that it has a response but it is not authenticated by the server and that the server is not the authority for the domain meaning it got the response from a different server.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

The response lists that jct.ac.il has two authorized certificate authorities and they are letsencrypt.org and sentigo.org

\*In my implementation of dig caa, I disregard answers with keyword iodef because that is about error reporting and not the actual CA

DNSMAP

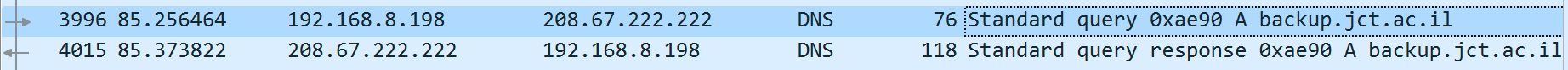
dnsmap is a tool that is used to try to identify any subdomains of a domain and their associated IP address. This is a helpful way to test if there are any vulnerable subdomains that are available to the public to access. It works by brute force querying tons of possible subdomain using either a built in list or a user provided list and prepending each entry in the list to given domain and sending DNS requests for each of them and returning all IP addresses associated with each successfully found subdomain.

Example of a sample of the responses given when doing dnsmap jct.ac.il:

A computer screen shot of a black screen

Description automatically generated

Here is an example of one of the queries- it is querying the subdomain backup.jct.ac.il.

This query is sent out and the subdomain actually exists so the response packet responds with the A record of the subdomain. Note how the error code in the response packet is set to 0, meaning there were no problems and the answer was successfully found.

A screenshot of a computer

Description automatically generated

In another query, for ba.jct.ac.il that domain does not exist so in the response packet, there was an error code 3 indicating that there is no such domain name that exists.

A screenshot of a computer screen

Description automatically generated

WHOIS

The whois command queries directly to specific whois servers. There are many different whois servers.A screenshot of a computer

Description automatically generated

Note how for 5 different whois queries, there are 6 different whois servers that are queried. (Because twice the client needs to send the query to a second server to receive the answer, and for Harvard and yale they use the same whois server.

There are different whois servers for the different tld’s so the client knows which whois server to query based on the tld of the domain. It sends a DNS query to the whois server belonging to tld of the given domain to get the IP address of that server, and then sends the whois query there. Sometimes that whois server contains all the information necessary for that domain and it ends there, but sometimes the whois server will send back the domain of another whois server that the client needs to query to get the complete information. In which case, the client will send another whois query to that second whois server. This process repeats until there are no other whois servers to query for that domain name.